

DYNAMISCHER LASTAUSGLEICH IN PARALLEL- RECHNERSYSTEMEN UNTER BERÜCKSICHTIGUNG DES WORK/EXCHANGE-MODELLS

Michael Witt and Wolfgang Weber

Kurzfassung. Aufgrund der Komplexität wissenschaftlicher Probleme gewinnt die Anwendung paralleler Rechensysteme immer mehr an Bedeutung. Um einen möglichst großen Gewinn an Rechenleistung zu erhalten, erfolgt die Zuordnung der Teilaufgaben unter Verwendung dynamischer Lastausgleichsverfahren. Die Effektivität eines Lastausgleiches wird dabei im wesentlichen durch den Kommunikationsaufwand bestimmt. Mit Hilfe des Work/Exchange-Modells wird durch analytische Behandlung der Zusammenhang zwischen Kommunikationsaufwand und Rechenleistung näher untersucht. Hieraus lassen sich Anforderungen sowohl an das Lastausgleichsverfahren als auch an die Hardware des Multiprozessorsystems ableiten, welche die Effizienz eines Lastausgleichs wesentlich erhöhen.

1. Einleitung

In vielen wissenschaftlichen Bereichen gibt es Problemstellungen, welche die Kapazität konventioneller Rechensysteme bei weitem übersteigen. Als Beispiele seien

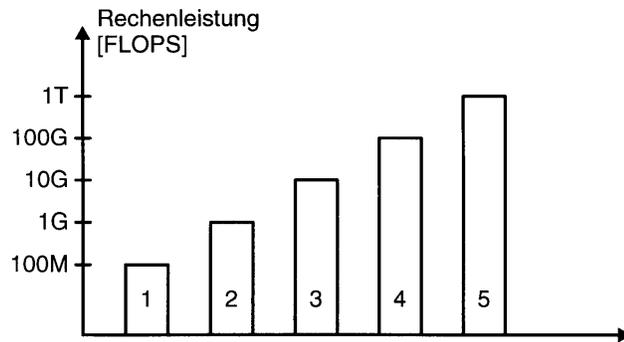
- die Klimaforschung
- die Genforschung sowie
- die Strömungsmechanik genannt, um nur einige zu nennen. Eine Übersicht über die erforderliche Rechenleistung ist in Bild 1 dargestellt.

Da die Komplexität der Probleme in Zukunft weiter ansteigen wird, gewinnt die Hochleistungsrechentechnik immer mehr an Bedeutung. Bis zum Ende dieses Jahrzehnts werden Rechensysteme mit einer Leistung von mehr

Manuscript received February 10, 1995.

Dipl. Ing. M. Witt ist Wissenschaftlicher Mitarbeiter des Instituts für Datenverarbeitung Ruhr Universität Bochum.

Prof. Dr. Dr. hc. W. Weber ist Direktor des Instituts für Datenverarbeitung, Ruhr-Universität Bochum, 44780 Bochum, Deutschland.



1. Flugzeugkonstruktion
2. 48 Stunden Wettervorhersage, zweidimensionale Plasmamodellierung
3. 72 Stunden Wettervorhersage, dreidimensionale Plasmamodellierung, Fahrzeugkonstruktion
4. Thermodynamik, Molekularbiologie, Pharmazie
5. Globale Klimamodellierung, Strömungsdynamik, Gentechnik

Bild 1. Anforderungen an die Rechenleistung

als 1 Tera-Flop als notwendig angesehen [5], um die zukünftigen Anforderungen zu bewältigen. Die Weiterentwicklung der Prozessortechnologie erlaubt durch

- die Erhöhung der Taktfrequenz
- das Pipeling und
- die parallele Verarbeitung von Daten auf dem Prozessor

zwar eine Erhöhung der Rechenleistung, jedoch sind die physikalischen Grenzen schon absehbar. So kann sich ein Signal auf dem Prozessorchip maximal mit Lichtgeschwindigkeit ausbreiten. In der Praxis sind jedoch nur Signallaufzeiten in der Größenordnung $10^7 m/sec$ realisierbar, so daß sich unter Berücksichtigung von Laufzeiteffekten als oberster Grenzwert eine Rechenleistung ergibt, die unterhalb von 1 Giga-Flop liegt.

1.1 Parallele Rechensysteme zur Lösung komplexer Probleme

Da aus den genannten Gründen ein sequentieller Rechner diese Grenze nicht überschreiten kann, werden zur Lösung komplexer Problemstellungen zunehmend parallele Rechensysteme eingesetzt. Diese bieten neben einer hohen Rechenleistung einige weitere Vorteile. Als wesentlich sind dabei

- die geringen Kosten,

- die leichte Erweiterbarkeit und
- der hohe Verfügbarkeitsgrad

eines derartigen Systems zu nennen. Ein Gewinn an Rechenleistung (Speedup) setzt jedoch voraus, daß die Aufgabenstellung weitestgehend parallelisiert werden kann. Nur unter dieser Voraussetzung ist der nach Amdahl [1] nicht zu vermeidende sequentielle Anteil eines Programmes vernachlässigbar. Mathematisch wird dieser Zusammenhang beschrieben durch

$$S_p = \frac{t_{s,g}}{t_s + \frac{t_p}{N}} \quad (1)$$

wobei N der Prozessoranzahl und $t_{s,g}$ der Ausführungszeit in einem sequentiellen Rechner entspricht. Die Ausführungszeit in einem Parallelrechner-system hingegen ergibt sich definitionsgemäß als Summe von t_s und t_p/N , welche den sequentiellen bzw. parallelen Programmanteil repräsentieren.

Im Vergleich zu einem sequentiellen Rechner ergibt sich somit für ein N -Prozessorsystem ein theoretischer Speedup von N . Praktisch kann dieser Wert jedoch nicht erreicht werden, da störende Effekte, wie z. B. die Synchronisation zwischen den einzelnen Teilaufgaben und der für den Austausch von Daten erforderliche Kommunikationsaufwand in der obigen Gleichung nicht berücksichtigt werden.

Um eine maximale Rechenleistung zu erreichen, muß zudem eine geeignete Zuordnung der Teilaufgaben zu den einzelnen Rechenknoten erfolgen. Dieses stellt jedoch ein grundsätzliches Problem dar.

2. Methoden für den Lastausgleich

Ein Ansatz, dieses Problem zu lösen, ist eine möglichst gleichmäßige Verteilung der Rechenlast im System. Dieser Lösungsansatz wird im allgemeinen als Lastausgleich bezeichnet. Zur Realisierung werden dabei die unterschiedlichsten Verfahren verwendet, die zum größten Teil auf heuristischen Ansätzen beruhen. Derartige Ansätze werden hier jedoch nicht näher betrachtet. Allen Verfahren ist jedoch gemeinsam, daß sie verschiedene Einzelstrategien miteinander kombinieren, welche

- die Art, den Austausch und die Auswertung von Lastinformationen sowie
- die Verlagerung von Teilaufgaben im System

betreffen. Dabei entsteht neben dem Aufwand für die Datenkommunikation ein zusätzlicher Kommunikations- und Verwaltungsaufwand, der beim Entwurf eines Lastausgleichsverfahrens zu berücksichtigen ist.

2.1 Statische und dynamische Verfahren

Unter Berücksichtigung der genannten Einzelstrategien kann eine Klassifizierung der Verfahren vorgenommen werden. Die Klassifikationsmerkmale werden im weiteren kurz erläutert, ohne daß jedoch eine detaillierte Beschreibung der zugrundeliegenden Verfahren erfolgt.

Als statisch wird ein Verfahren bezeichnet, bei dem die Zuordnung der Teilaufgaben (a_i) zu den Prozessoren (p_j) unabhängig von der Laufzeit ist. Die Zuordnung wird dabei durch den Benutzer bzw. Compiler initiiert und kann z. B. in Form einer Tabelle erfolgen. Mathematisch kann dieser Zusammenhang durch eine eindeutige Korrespondenz zwischen den beiden Mengen $A = \{a_j\}$ und $P = \{p_j\}$ beschrieben werden.

Statische Verfahren haben jedoch entscheidende Nachteile. So ist ein a priori Wissen, insbesondere über die Laufzeiteigenschaften der Teilaufgaben, erforderlich. Es können jedoch maximal die Kommunikationsverbindungen in Form einer Matrix angegeben werden. Da die Mehrzahl der Probleme sich durch ein nichtdeterministisches Lastverhalten charakterisieren läßt, ist ein statisches Lastausgleichsverfahren somit in der Regel ungeeignet.

Eine Umverteilung der Teilaufgaben zur Laufzeit wird nur durch dynamische Verfahren unterstützt. Mathematisch läßt sich dies durch eine eindeutige Korrespondenz zwischen A und P beschreiben. Durch Anwendung eines dynamischen Verfahrens kann insbesondere eine Zuordnung unter Berücksichtigung der tatsächlichen Rechenlast erfolgen. Nachteilig ist jedoch der zusätzliche Kommunikationsaufwand, der durch die Weitergabe der ermittelten Lastzustände an eine zentrale bzw. verteilte Instanz hervorgerufen wird. Des weiteren entsteht durch die Auswertung der Lastinformationen ein zusätzlicher Verwaltungsaufwand.

2.2 Weitere Klassifikationsmerkmale

Die Unterteilung in statische und dynamische Verfahren ist das mit Abstand wichtigste Klassifikationsmerkmal. Darüberhinaus lassen sich noch weitere Klassifikationsmerkmale angeben, die, ohne auf Detailfragen näher einzugehen, im folgenden erläutert werden:

- *optimal vs. suboptimal*: Die Anwendung optimaler Verfahren führt immer zu einer optimalen Kostenfunktion. Dies natürlich nur unter der Bedingung, daß alle Informationen über den Systemzustand und den Betriebsmittelbedarf vollständig zur Verfügung stehen. Alle anderen Verfahren führen zu suboptimalen Lösungen, wobei sich eine Unterteilung in approximative und heuristische Ansätze vornehmen läßt.

Bei den approximativen Verfahren wird der Lösungsraum einschränkt und nach einem lokalen Optimum der Kostenfunktion gesucht. Im Gegensatz hierzu gehen heuristische Verfahren von vermuteten Zusammenhängen zwischen den optimierenden und den anderen Größen aus. Hiermit verbunden ist der Nachteil, daß der Zusammenhang mit der Maximierung der Rechenleistung nur intuitiv begründbar ist und sich nur experimentell beweisen läßt.

- *zentral vs. dezentral*: Ein weiteres Klassifikationsmerkmal läßt sich aus der Instanz, welche die Entscheidung für die Zuordnung der einzelnen Teilaufgaben trifft, ableiten. Bei den zentralen Verfahren wird diese Entscheidung von einem einzelnen Prozessor getroffen. Nachteilig ist dabei, daß dieser Knoten sehr schnell zum Nadelöhr des Systems werden kann. Im Gegensatz hierzu liegt den dezentralen Verfahren eine verteilte Entscheidungsinstanz zugrunde. Eine Teilinstanz ist dabei für einen bestimmten Rechenknoten oder für eine Teilmenge von Prozessoren verantwortlich.
- *kooperativ vs. autonom*: Bei den autonomen Verfahren wird der Zustand des Prozessornetzes nicht berücksichtigt. Die Zuteilung der Teilaufgaben erfolgt aufgrund fixen Wissens über die Topologie und Rechenleistung anderer Rechenknoten. Im Gegensatz hierzu kann durch die Anwendung kooperativer Verfahren der Zustand der unmittelbaren Nachbarn oder sogar des gesamten Netzes berücksichtigt werden.
- *senderinitiiert vs. empfangeninitiiert*: Senderinitiierte Verfahren (z.B. Bidding-Algorithmus) zeichnen sich dadurch aus, daß die stark belasteten Knoten versuchen, Teilaufgaben auf andere weniger belastete Rechenknoten zu verlagern. Hiermit ist jedoch ein entscheidender Nachteil verbunden. Die ohnehin schon stark belasteten Knoten werden durch den zusätzlichen Kommunikationsaufwand noch stärker belastet. Von Vorteil ist jedoch, daß keine unnötigen Migrationsversuche erfolgen. Ein senderinitiiertes Lastausgleich ist somit nur bei durchschnittlich geringer Systemlast sinnvoll.

Empfängerinitiierte Verfahren (z.B. Drafting-Algorithmus) hingegen sind dadurch gekennzeichnet, daß sich unbelastete Knoten direkt zu erkennen geben, wodurch sich der Kommunikationsaufwand reduzieren läßt. Da es jedoch zu unnötigen Migrationsversuchen kommen kann, ist dieses Verfahren nur für hohe Systemlasten geeignet. Aufgrund der genannten Vor- und Nachteile ist eine Kombination von sender- und empfangeninitiiertem Lastausgleich sinnvoll. Zur Auswahl der Lastverteilungsstrategie werden Schwellwerte verwendet, die auf der Rechenlast

des Systems basieren. Als Beispiel für ein derartiges Verfahren ist der Dipstick-Algorithmus zu nennen.

- *adaptiv vs. nichtadaptiv*: Bei den adaptiven Verfahren erfolgt eine Anpassung der Entscheidungsparameter unter Berücksichtigung vorangegangener Entscheidungen, wohingegen nichtadaptive Verfahren auf festen Parametern und einer festen Zuordnungsstrategie basieren.

3. Das Work-Exchange-Modell

Für den Entwurf eines Lastausgleichsverfahrens ist der benötigte Aufwand von entscheidender Bedeutung. Insbesondere stellt sich die Frage, inwieweit ein Zusammenhang zwischen dem zusätzlichen Kommunikationsaufwand und der Rechenleistung des Systems besteht. Das Work/Exchange-Modell stellt dabei einen Versuch dar, durch analytische Behandlung des Problems qualitative Aussagen zu gewinnen.

3.1 Annahmen

Zunächst geht das Work/Exchange-Modell von einem 2-Prozessor System aus. Eine Herleitung analytischer Gleichungen ist jedoch nur unter bestimmten stark vereinfachenden Annahmen möglich [7]. Diese werden im weiteren näher erläutert:

- Auf beiden Prozessoren werden Zyklen abgearbeitet, die sich aus einem Berechnungs- und einem Kommunikationsanteil zusammensetzen.
- Die eigentliche Rechenlast der Prozessoren kann dabei unterschiedlich sein, ist jedoch innerhalb eines Zyklus konstant.
- Der Austausch von Informationen basiert auf einem nichtblockierenden SEND und einem blockierenden READ.
- Die Zeit für den Verbindungsaufbau ist für die Dauer des Information-saustausches entscheidend.

3.2 Ergebnisse

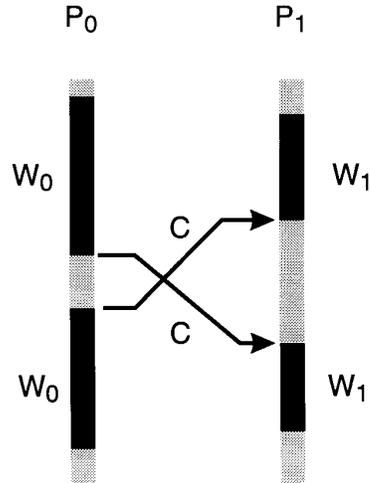
Mit den genannten Annahmen lassen sich für den Zyklus k die Ausführungs- ($t_{j,k}$) und die Wartezeit ($i_{j,k}$) des jeweiligen Prozessors bestimmen. Eine Wartezeit tritt in dem stark vereinfachten Modell nur dann auf, wenn ein Prozessor auf eintreffende Nachrichten wartet.

Im ausgeglichenem Zustand senden und empfangen beide Prozessoren zur gleichen Zeit, d.h. der Zyklus der Prozessoren ist identisch. Die Ausführungs-

rungszeit (t_{k-}) ergibt sich dabei als

$$t_{k-} = k_u \left(\frac{W_0 + W_1}{2} + C \right),$$

wobei, wie in Bild 2 dargestellt, W_j der Arbeitslast des Prozessors p_j und C dem Kommunikationsaufwand entspricht.



P_i Prozessor x ($i = 0, 1$)

W_i Arbeitslast von Prozessor i ($i = 0, 1$)

$D = |W_0 - W_1|$ Differenz der Arbeitslasten

C Kommunikationskosten

Bild 2. Veranschaulichung des Work/Exchange-Modells

Aus dem Verhältnis der beiden Ausführungszeiten

$$S_{p,k} = \frac{t_{k-}}{\max_{j=0,1} t_{j,k}}$$

läßt sich eine Aussage über den Speedup, der sich durch eine Umverteilung von Teilaufgaben erreichen läßt, treffen. Die entsprechenden Gleichungen sind in Tabelle 1 angegeben.

Wenn k gegen unendlich strebt, ergibt sich ein Grenzwert, der jedoch vom Speedup in den ersten Zyklen stark abweicht. Zu diesem Zeitpunkt ist somit eine Aussage über die Effektivität eines Lastausgleichs nicht möglich.

Tabelle 1. Speedup-Gleichungen für das 2-Prozessor-Modell

	$0 \leq C \leq D/2$	$D/2 < C \leq D$	$C \geq D$
$S_{p,k}$ $k = 1, 3, 5, \dots$	$1 + \frac{D - \frac{j-1}{j}2C}{2(W_0 + C) - D}$	$1 + \frac{D - C}{k[2(W_0 + C) - D]}$	1
$S_{p,k}$ $k = 2, 4, 6, \dots$		$1 + \frac{D}{k[2(W_0 + C) - D]}$	$1 + \frac{D}{k[2(W_0 + C) - D]}$
$S_{Max,k}$ $k = 1, 3, 5, \dots$	2	$1 + \frac{1}{2k}$	1
$S_{Max,k}$ $k = 2, 4, 6, \dots$			$1 + \frac{1}{3k}$

Des weiteren ist der Kommunikationsaufwand für den Gewinn an Rechenleistung entscheidend. Dies bedeutet, daß Lastausgleichsverfahren, die Kommunikationskosten und Zugriffskonflikte ignorieren, ungeeignet sind.

Die Gleichungen stehen zudem im Widerspruch zu der Vermutung, daß ein Lastausgleich in allen Fällen und zu jedem Zeitpunkt zu einer Steigerung der Rechenleistung führt. Grundsätzlich ist jedoch ein Lastausgleich in Parallelrechnersystemen sinnvoll.

3.3 Verallgemeinerung des Modells

Bisher wurde von einer periodischen Auswertung der Systemlast ausgegangen. Ein derartiger Ansatz ist jedoch nur für Problemstellungen mit deterministischem Lastungleichgewicht geeignet. Die meisten Probleme, wie z.B. die Wettervorherage, sind jedoch durch ein eher stochastisches Lastverhalten bestimmt, so daß sich zunächst die Frage nach einer Erweiterung des Verfahrens stellt. Diese besteht zum einen darin, daß einerseits die Initiierung des Lastausgleichs entscheidungsabhängig ist und andererseits die Auswertung der Systemlast in keinem festen Zeitintervall erfolgt.

Um die Auswertung der Lastzustände zeitvariabel zu halten, ist eine Vorhersage der Rechenzeit ohne bzw. mit Lastausgleich erforderlich. Für

eine analytische Behandlung des Problems stellt sich nunmehr die Frage, unter welchen Voraussetzungen sich eine Voraussage treffen läßt. Eine genaue Betrachtung zeigt, daß unter der Annahme der temporären Lokalität die Herleitung analytischer Gleichungen möglich ist.

Die temporäre Lokalität ist dabei definiert als

$$W_{j,k+1} = W_{j,k} \pm \Phi_{j,k} \text{ mit } \Phi_{j,k} \ll W_{j,k}.$$

Dies bedeutet, daß eine hinreichende Stetigkeit der Systemlast vorausgesetzt wird. Die meisten wissenschaftlichen Probleme lassen sich durch ein solches Verhalten charakterisieren [7].

Nur unter dieser Voraussetzung lassen sich aus dem vorangegangenen Zyklus Näherungen für den Kommunikationsaufwand C und für die Arbeitslast $W_{i,k}$ bestimmen, mit denen eine Voraussage der Ausführungszeit ohne $(t_{j,k})$ bzw. mit Lastausgleich (t_{k-}) möglich ist. Die Initiierung des Lastausgleichs erfolgt unter Berücksichtigung des folgenden Kriteriums:

- Wenn $t_{k-} + C_- \leq t_{j,k}$, dann führe einen Lastausgleich durch.
- Wenn $t_{k-} + C_- > t_{j,k}$, dann führe keinen Lastausgleich durch.

Mit C_- werden dabei die Kosten für den eigentlichen Lastausgleich berücksichtigt.

Bisher wurde der Aufwand für die Schätzung des Kommunikation- aufwandes und der Arbeitslast, hier als C_S bezeichnet, vernachlässigt. Dies läßt sich damit begründen, daß $t_{j,k}$, t_{k-} sowie C_- selbst Schätzungen sind und C_S konstant ist. Für die Änderung des Auswertungsintervalls ist C_S jedoch von größerer Bedeutung. So erfolgt eine Anpassung dann, wenn

$$t_{k-} + C_- + C_S + \alpha \leq t_{j,k} \text{ (Verringerung der Zyklusdauer)}$$

oder

$$t_{k-} + C_- + C_S + \alpha \leq t_{j,k} \text{ (Erhöhung der Zyklusdauer)}.$$

Ansonsten bleibt das Auswertungsintervall unverändert. Des weiteren erfolgt keine Anpassung, wenn beide Seiten der Ungleichungen näherungsweise gleich sind. Dies wird durch die Korrekturfaktoren α und β sichergestellt.

Im Sinne einer weiteren Verallgemeinerung des Modells stellt sich nunmehr die Frage, inwieweit sich dieses auf ein N-Prozessor- Modell erweitern läßt. Aufgrund der Komplexität des Problems kann eine Herleitung analytischer Gleichungen nur unter den Voraussetzungen, daß

- die Ausführungszeit jeder Teilaufgabe gleich ist,
- die Kommunikationskosten von Prozessor zu Prozessor konstant sind und
- das verbindende Kommunikationsnetz Rundsprucheeigenschaften besitzt, erfolgen. Eine Übersicht über den erreichbaren Speedup ist in Tabelle 2 dargestellt.

Tabelle 2. Speedup-Gleichungen für das N-Prozessor-Modell.

$$D_{2-} = W_{01} - W_-$$

$$H_{01} = \frac{D_{01}}{2} = \frac{(W_0 - W_1)}{2} \quad (\text{Annahme: } W_0 > W_1 > W_j \text{ mit } j = 2, \dots, N)$$

$$W_- = \frac{1}{N} \sum_{j=0}^{N-1} W_j$$

$$C_B = H_{01} - C_{ref} \quad (C_{ref} : \text{Referenzgröße})$$

	$0 \leq C \leq H_{01}$	$0 \leq C \leq 2H_{01}$	$C \geq 2H_{01}$
$S_{p,k}$ $k = 1, 3, 5, \dots$	$1 + \frac{D_{2-} + C_B + \frac{1}{k}C}{W_- + C}$	$1 + \frac{D_{2-} + \frac{1}{k}H_{01}}{W_- + C}$	$1 + \frac{D_{2-} + \frac{1}{k}H_{01}}{W_- + C}$
$S_{p,k}$ $k = 2, 4, 6, \dots$		$1 + \frac{D_{2-} + \frac{1}{k}C_B}{W_- + C}$	$1 + \frac{D_{2-}}{W_- + C}$
$\lim_{k \rightarrow \infty} S_{p,k}$	$1 + \frac{D_{2-} + C_B}{W_- + C}$	$1 + \frac{D_{2-}}{W_- + C}$	

Mit dem erhaltenen Modell lassen somit sich ähnliche qualitative Aussagen über den Einfluß des Kommunikationsaufwandes und über das Lastausgleichsverfahren treffen.

4. Schlußfolgerungen

Die bisherigen Betrachtungen zeigten, daß ein Lastausgleich nicht in allen Fällen zur Steigerung der Rechenleistung sinnvoll ist. Die Effektivität

eines Lastausgleichs wird dabei durch den Kommunikationsaufwand, der aufgrund des Austausches von Daten und der Systemverwaltung entsteht, bestimmt. Bei der Konzipierung eines Lastausgleichsverfahrens ist somit die Minimierung des Kommunikationsaufwandes von entscheidender Bedeutung.

4.1 Anforderungen an das Verfahren

Eine vollständige Umsetzung der mit Hilfe des Modells gewonnenen Erkenntnisse ist schwierig, da schon die Auswahl für den Allgemeinfall geeigneter Schätzer ein grundlegendes Problem darstellt. Es bleibt jedoch festzuhalten, daß für ein N-Prozessor-System ohne Rundsprucheeigenschaften eine analytische Behandlung nicht möglich ist. Dies bedeutet, daß ein praktischer Ansatz eines Lastausgleichsverfahrens nur über ein konkretes Multiprozessorsystem erfolgen kann. An das Verfahren stellen sich dabei folgende Anforderungen:

- Ein Lastausgleich sollte unter Berücksichtigung des aktuellen Systemzustandes und der entstehenden Kosten initiiert werden.
- Die Auswertung der Lastzustände sollte in keinem festen Zeitintervall, sondern auf der Basis der tatsächlichen Systemlast erfolgen.

4.2 Anforderungen an die Hardware

Wie bereits erwähnt, ist für die Effektivität des Lastausgleichsverfahrens die Minimierung des Kommunikations- und Verwaltungsaufwandes entscheidend. Dies kann nicht jedoch nur durch das Verfahren erfolgen, sondern muß von der Hardware entsprechend unterstützt werden.

Insbesondere ist eine Trennung von Aufgabenbearbeitung und Systemverwaltung sinnvoll. Ein Ansatz zur Lösung des Problems besteht, wie in Bild 3 dargestellt, darin, die Systemtopologie auf zwei Ebenen zu erweitern.

In der Ausführungsebene erfolgt die Bearbeitung der Teilaufgaben, wohingegen die Verwaltungsebene ausschließlich der Systemverwaltung dient und somit dem Benutzer verborgen bleibt. Im einfachsten Fall kann der Verwaltungs- und der Ausführungsebene je 1 Prozessor pro Rechenknoten zugeordnet werden. Über einen längeren Zeitraum gesehen, ist der Verwaltungsaufwand jedoch durchaus geringer, so daß aus diesem Grund eine Zusammenfassung von Prozessoren in der Ausführungsebene sinnvoll erscheint.

Unter der Voraussetzung, daß die Kommunikation zwischen der Verwaltungs- und Ausführungsebene über spezielle Kommunikationskanäle erfolgt,

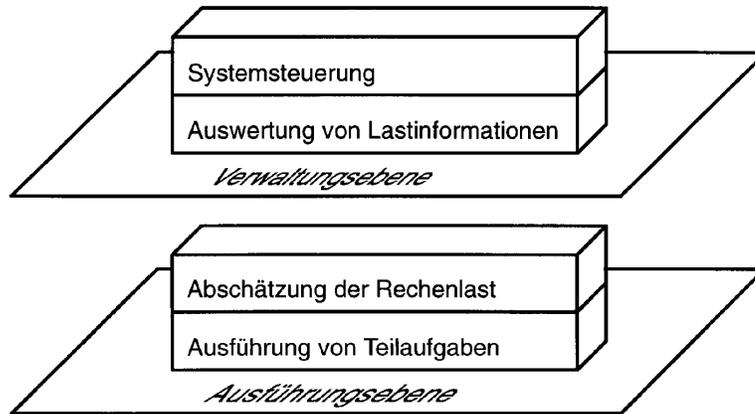


Bild 3. Ebenen der erweiterten Systemtopologie.

betrifft die Kommunikation zwischen den Prozessoren in der Ausführungsebene fast ausschließlich die Datenkommunikation. Eine weitere Reduzierung des Kommunikationsaufwandes ist durch ein dynamisches frei konfigurierbares Kommunikationsnetz in der Ausführungsebene möglich.

Zur Zeit entsteht am Lehrstuhl für Datenverarbeitung ein Parallelrechnersystem auf Transputerbasis, welchen den genannten Anforderungen genügt. Der Aufbau eines Rechenknotens ist in Bild 4 dargestellt.

Als Verwaltungsprozessor wird dabei ein T805 pro Rechenknoten verwendet. Die einzelnen Teilaufgaben werden in 4 T9000-kompatiblen Subtransputermodulen, die über den Kreuzschienenverteiler C104 miteinander verbunden sind, ausgeführt. Des weiteren wird der Austausch von Lastinformationen über 4 Dual-Ported-RAM, die sich zwischen den Subtransputern und dem Transputer auf der Verwaltungsebene, hier als Kontrolltransputer bezeichnet, befinden, als Rundspruch durchgeführt. Die Zuordnung der einzelnen Teilaufgaben hingegen erfolgt über eine zusätzliche Link-Schnittstelle, die direkt mit dem C104 verbunden ist. Da die Link-Protokolle des T9000 und des T805 unterschiedlich sind, wird die Schnittstelle mittels des Protokollkonverters C100 entsprechend angepaßt.

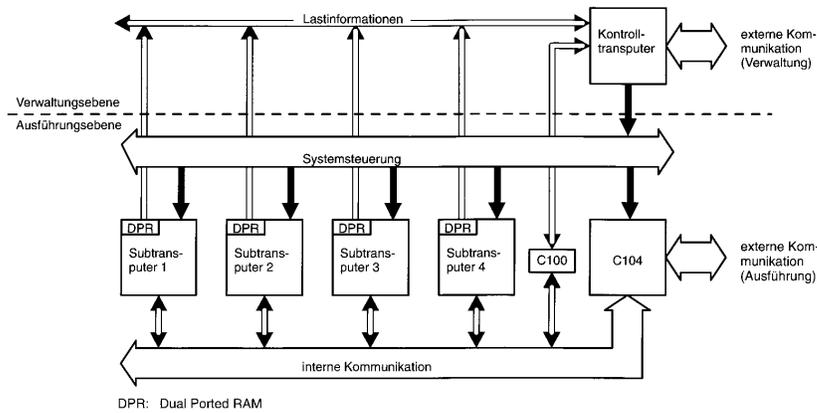


Bild 4. Aufbau eines Rechenknotens

REFERENCES

1. AMDAHL, G. M.: *Validity of the Single-Processor Approach to Achieving Large-Scale Computing Capabilities*. Proceedings of the AFIPS Conference, Vol. 30, New York, 1967.
2. CHOU, T. C. K. AND ABRAHAM, J. A.: *Load Balancing in Distributed Systems*. IEEE Transactions On Software Engineering, vol. SE-8, No. 4, July 1992.
3. HARMS, U.: *Amdahls Gesetz und die Parallelverarbeitung*. Design & Elektronik, Nr. 26, 1990.
4. HE, X.: *Eine Übersicht über die Lastverteilung in verteilten Systemen*. Interner Bericht Universität Kaiserslautern, Fachbereich Informatik, Mai 1989.
5. RUBBIA, C. U. A.: *Report of the EEC Working Group on High-Performance Computing*. Europäische Gemeinschaft, Brüssel, 1991.
6. SCHABERNACK, J.: *Lastenausgleichsverfahren in verteilten Systemen. Überblick und Klassifikation*. Informationstechnik, Nr. 5, 1992.
7. WIKSTROM, M. C.: *The Work/Exchange Model: A generalized approach to dynamic load balancing*. Dissertation, Iowa States University, 1991.